

Simulator 驱动程序

© 2021 PTC Inc. 保留所有权利。

目录

Simulator 驱动程序	1
目录	2
Simulator 驱动程序	3
概述	3
设置	3
通道属性 - 常规	4
通道属性 - 写入优化	4
通道属性 - 高级	5
通道属性 - 持久化	5
通道属性 - Configuration API	6
设备属性 - 常规	8
操作模式	8
设备属性 - 扫描模式	9
设备属性 - 标记生成	9
设备属性 - Configuration API	11
数据类型说明	13
地址说明	15
8 位设备地址	15
16 位设备地址	15
模拟函数	16
斜坡函数	16
随机函数	17
正弦函数	17
User 函数	17
事件日志消息	19
无法加载项状态数据。原因: <原因>。	19
无法保存项状态数据。原因: <原因>。	19
索引	20

Simulator 驱动程序

帮助版本 1.044

目录

[概述](#)

什么是 Simulator 驱动程序？

[设置](#)

如何配置使用此驱动程序的设备？

[数据类型说明](#)

哪种数据类型可用于模拟设备？

[地址说明](#)

如何在模拟设备上指定地址？

[事件日志消息](#)

Simulator 驱动程序会产生哪些错误消息？

概述

Simulator 驱动程序提供将 Simulator 设备连接到 OPC 客户端应用程序的可靠方式；其中包括 HMI、SCADA、Historian、MES、ERP 和无数自定义应用程序。其用于测试 OPC 服务器软件产品，而无需外部设备。

设置

支持的设备

8 位
16 位

◆ **注意：**除了 1000 可变长度、读/写字符串位置，每个设备最多还可支持 10000 个可寻址读/写寄存器和常量位置。有关详细信息，请参阅[地址说明](#)。

通道和设备限制

此驱动程序支持的最大通道数量为 1024。此驱动程序所支持设备的最大数量为每通道 999 个。

设备 ID

可以为仿真器设备分配设备 ID，范围为 1 到 999。在同一通道内使用不同的模型类型时，需要使用唯一的设备 ID。

实时数据模拟

驱动程序通过增加每次读取为整数数据类型的寄存器数据来模拟实时数据。除了模拟类似于 PLC 的设备的基于寄存器的简单内存之外，Simulator 驱动程序还支持四个高级模拟函数。这些新的模拟函数包括 RAMP、SINE、RANDOM 和用户定义函数。

每个新模拟标记的结构类似编程语言中的函数调用。使用每个函数要求应用适当的属性确定所需的模拟效果。RAMP 函数，可用于设置更改率、下限、上限和增量值。SINE 函数，可用于设置更改率、下限、上限、频率和相位。RANDOM 函数，可用于设置更改率、下限和上限。但是，新模拟函数最具创造性的是其为“用户定义”函数。

“用户定义”函数同样可用于指定更改率。与 RAMP、SINE、RANDOM 的预设模拟输出不同，“用户定义”函数可用于创建数据序列。在上限或下限位置，“用户定义”函数接受以逗号分隔的项列表。项列表可以是数字数据或字符串数据。列表建立后，“用户定义”函数将按照指定速率循环列表中的元素。“用户定义”函数可创建可用于镜像实际输出和结果的复杂演示。

◆ 有关详细信息，请参阅[模拟函数](#)。

通道属性 - 常规

此服务器支持同时使用多个通信驱动程序。服务器项目中使用的各个协议或驱动程序称为通道。服务器项目可以由具有相同通信驱动程序或具有唯一通信驱动程序的多个通道组成。通道充当 OPC 链路的基础构建块。此组用于指定常规通道属性，如标识属性和操作模式。

属性组	<input type="checkbox"/> 标识	
常规	名称	通道 1
写优化	说明	
高级	驱动程序	Simulator
持久存储	<input type="checkbox"/> 诊断	
	诊断数据捕获	禁用

标识

“名称”: 指定此通道的用户定义标识。在每个服务器项目中，每个通道名称都必须是唯一的。尽管名称最多可包含 256 个字符，但在浏览 OPC 服务器的标记空间时，一些客户端应用程序的显示窗口可能不够大。通道名称是 OPC 浏览器信息的一部分。该属性是创建通道所必需的。

● 有关保留字符的信息，请参阅服务器帮助中的“如何正确命名通道、设备、标记和标记组”。

“说明”: 指定此通道的用户定义信息。

● 在这些属性中，有很多属性 (包括“说明”) 具有关联的系统标记。

“驱动程序”: 为该通道指定的协议/驱动程序。该属性指定在通道创建期间选择的设备驱动程序。它在通道属性中为禁用设置。该属性是创建通道所必需的。

● **注意**: 服务器全天在线运行时，可以随时更改这些属性。其中包括更改通道名称以防止客户端向服务器注册数据。如果客户端在通道名称更改之前已从服务器中获取了项，那么这些项不会受到任何影响。如果客户端应用程序在通道名称更改之后发布项，并尝试通过原来的通道名称重新获取项，则该项将不被接受。考虑到这一点，一旦开发完成大型客户端应用程序，就不应对属性进行任何更改。采用适当的用户角色和权限管理来防止操作员更改属性或访问服务器功能。

诊断

“诊断数据捕获”: 启用此选项后，通道的诊断信息即可提供给 OPC 应用程序。由于服务器的诊断功能所需的开销处理量最少，因此建议在需要时使用这些功能，而在不需要时禁用这些功能。默认设置为禁用状态。

● **注意**: 如果驱动程序不支持诊断，则该属性不可用。

● 有关详细信息，请参阅服务器帮助中的“通信诊断”和“统计信息标记”。

通道属性 - 写入优化

服务器必须确保从客户端应用程序写入的数据能够准时发送到设备。为此，服务器提供了优化属性，用以满足特定需求或提高应用程序响应能力。

属性组	<input type="checkbox"/> 写优化	
常规	优化方法	仅写入所有标记的最新值
写优化	占空比	10
高级		
持久存储		

写入优化

“优化方法”: 控制如何将写入数据传递至底层通信驱动程序。选项包括：

- **“写入所有标记的所有值”**: 此选项可强制服务器尝试将每个值均写入控制器。在此模式下，服务器将持续收集写入请求并将它们添加到服务器的内部写入队列。服务器将对写入队列进行处理并尝试通过将数据尽快写入设备来将其清空。此模式可确保从客户端应用程序写入的所有数据均可发送至目标设备。如果写入操作顺序或写入项的内容必须且仅能显示于目标设备上，则应选择此模式。

- **“写入非布尔标记的最新值”**：由于将数据实际发送至设备需要一段时间，因此对同一个值的多次连续写入会存留于写入队列中。如果服务器要更新已位于写入队列中的某个写入值，则需要大大减少写入操作才能获得相同的最终输出值。这样一来，便不会再有额外的写入数据存留于服务器队列中。几乎就在用户停止移动滑动开关时，设备中的值达到其正确值。根据此模式的规定，任何非布尔值都会在服务器的内部写入队列中更新，并在下一个可能的时机发送至设备。这可以大大提高应用性能。
 - **注意**：该选项不会尝试优化布尔值的写入。它允许用户在不影响布尔运算的情况下优化 HMI 数据的操作，例如瞬时型按钮等。
- **“写入所有标记的最新值”**：该选项采用的是第二优化模式背后的理论并将其应用至所有标记。如果应用程序只需向设备发送最新值，则该选项尤为适用。此模式会通过当前写入队列中的标记发送前对其进行更新来优化所有的写入操作。此为默认模式。

“占空比”(Duty Cycle)：用于控制写操作与读操作的比率。该比率始终基于每一到十次写入操作对应一次读取操作。占空比的默认设置为 10，这意味着每次读取操作对应十次写入操作。即使在应用程序执行大量的连续写入操作时，也必须确保足够的读取数据处理时间。如果将占空比设置为 1，则每次读取操作对应一次写入操作。如果未执行任何写入操作，则会连续处理读取操作。相对于更加均衡的读写数据流而言，该特点使得应用程序的优化可通过连续的写入操作来实现。

● **注意**：建议在将应用程序投入生产环境前使其与写入优化增强功能相兼容。

通道属性 - 高级

此组用于指定高级通道属性。并非所有驱动程序都支持所有属性，因此不会针对不支持的设备显示“高级”组。

属性组	<input type="checkbox"/> 非规范浮点数处理	
常规	浮点值	替换为零
以太网通信	<input type="checkbox"/> 设备间延迟	
写优化	设备间延迟 (毫秒)	0
高级		
通信序列化		

“非规范浮点数处理”：非规范值定义为无穷大、非数字 (NaN) 或不正规编号。默认值为“替换为零”。具有原生浮点数处理功能的驱动程序可能会默认设置为“未修改”。通过非规范浮点数处理，用户可以指定驱动程序处理非规范 IEEE-754 浮点数据的方式。选项说明如下：

- **“替换为零”**：此选项允许驱动程序在将非规范 IEEE-754 浮点值传输到客户端之前，将其替换为零。
- **“未修改”**：此选项允许驱动程序向客户端传输 IEEE-754 不正规、规范、非数字和无穷大值，而不进行任何转换或更改。

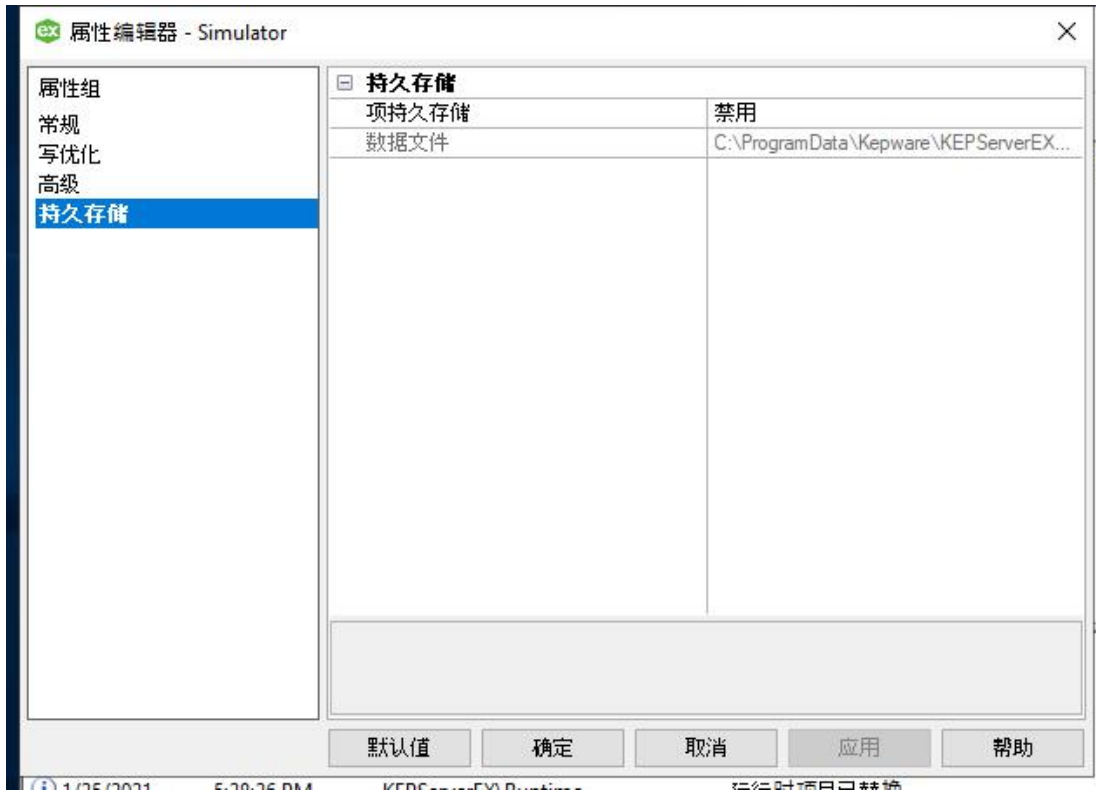
● **注意**：如果驱动程序不支持浮点值或仅支持所显示的选项，则此属性不可用。根据通道的浮点规范化设置，将仅对实时驱动程序标记 (如值和数组) 进行浮点规范化。例如，此设置不会影响 EFM 数据。

● 有关浮点值的详细信息，请参阅服务器帮助中的“如何使用非规范化浮点值”。

“设备间延迟”：指定在接收到同一通道上的当前设备发出的数据后，通信通道向下一设备发送新请求前等待的时间。设置为零 (0) 将禁用延迟。

● **注意**：此属性并不适用于所有驱动程序、型号和相关设置。

通道属性 - 持久化



“项目暂留”：选择“启用”可将设备配置为在运行期间保留其 **K** (常量)、**R** (寄存器) 和 **S** (字符串) 地址的值。当前不支持模拟函数的项目暂留。该选项启用后，当服务器项目关闭 (或服务器关闭) 时，存储在通道上所配置的所有设备的 **K**、**R** 和 **S** 地址中的值都将保存到文件。当下次打开服务器项目时，这些值可从文件中恢复。使用此功能时，每个通道都会使用一个单独的文件。默认设置为禁用状态。

● 有关详细信息，请参阅 [模拟函数](#)。

“数据文件”：指定此通道设备数据的存储位置。此时需要一个完全限定的路径和文件名。驱动程序将在其路径中创建文件和文件夹，但用户必须使用该功能每个“模拟器”通道指定一个唯一的文件。

通道属性 - Configuration API

以下属性使用 Configuration API 服务来定义通道。

常规属性

`common.ALLTYPES_NAME` * 必需的参数

● **注意**：更改此属性将导致 API 端点 URL 发生改变。

`common.ALLTYPES_DESCRIPTION`

`servermain.MULTIPLE_TYPES_DEVICE_DRIVER` * 必需的参数

`servermain.CHANNEL_DIAGNOSTICS_CAPTURE`

以太网通信属性

`servermain.CHANNEL_ETHERNET_COMMUNICATIONS_NETWORK_ADAPTER_STRING`

高级属性

`servermain.CHANNEL_NON_NORMALIZED_FLOATING_POINT_HANDLING`

写入优化

servermain.CHANNEL_WRITE_OPTIMIZATIONS_METHOD

servermain.CHANNEL_WRITE_OPTIMIZATIONS_DUTY_CYCLE

● **另请参阅:** 服务器帮助系统“*Configuration API* 服务”部分。

设备属性 - 常规

一个设备代表通信通道上的单一目标。如果驱动程序支持多个控制器，则用户必须为每个控制器输入一个设备 ID。

Property Groups	<input checked="" type="checkbox"/> Identification	
General	Name	
Scan Mode	Description	
	Channel Assignment	
	Driver	
	Model	
	ID Format	Decimal
	ID	2

标识

“名称”: 指定设备的名称。此为用户定义的逻辑名称，最长可达 256 个字符，并且可以用于多个通道。

● **注意**: 尽管描述性名称通常是不错的选择，但浏览 OPC 服务器的标记空间时，一些 OPC 客户端应用程序的显示窗口可能不够大。设备名称和通道名称也成为浏览树信息的一部分。OPC 客户端中，通道名称和设备名称的组合将显示为“通道名称.设备名称”。

● 有关详细信息，请参阅服务器帮助中的“如何为通道、设备、标记和标记组正确命名”。

“说明”: 指定此设备的用户定义信息。

● 在这些属性中，有很多属性 (包括“说明”) 具有关联的系统标记。

“通道分配”: 指定该设备当前所属通道的用户定义名称。

驱动程序: 为该设备选择的协议驱动程序。

“型号”: 指定与此 ID 关联的设备的类型。下拉菜单中的内容取决于正在使用的通信驱动程序类型。驱动程序不支持的型号将被禁用。如果通信驱动程序支持多个设备型号，则只有当设备未与任何客户端应用程序连接时，才能改变型号的选择。

● **注意**: 如果通信驱动程序支持多种模型，则用户应将模型选择与物理设备进行匹配。如果下拉列表菜单中未显示该设备，则选择与目标设备最相近的模型。一些驱动程序支持名为“开放式”的型号选择，该选择使用户无需了解目标设备的具体信息即可进行通信。有关详细信息，请参阅驱动程序帮助文档。

ID: 指定设备驱动程序特定的工作站或节点。输入的 ID 类型取决于正在使用的通信驱动程序。对于许多通信驱动程序而言，ID 是一个数值。支持数字 ID 的驱动程序使用户能够输入格式可更改的数值，以适应应用需要或所选通信驱动程序特点。默认情况下，该格式由驱动程序设置。选项包括十进制、八进制和十六进制。

● **注意**: 如果驱动程序基于以太网，或者支持非常规工作站或节点名称，则可使用设备的 TCP/IP 地址作为设备 ID。TCP/IP 地址包含四个由句点分隔的值，每个值的范围在 0 至 255 之间。某些设备 ID 基于字符串。根据不同驱动程序，也可以在 ID 字段中配置其他属性。有关详细信息，请参阅驱动程序的帮助文档。

操作模式

Property Groups	<input checked="" type="checkbox"/> Identification	
General	<input checked="" type="checkbox"/> Operating Mode	
Scan Mode	Data Collection	Enable
	Simulated	No

数据收集: 此属性控制设备的活动状态。尽管默认情况下会启用设备通信，但可使用此属性禁用物理设备。设备处于禁用状态时，不会尝试进行通信。从客户端的角度来看，数据将标记为无效，且不接受写入操作。通过此属性或设备系统标记可随时更改此属性。

“模拟”: 使设备进入或退出模拟模式。在此模式下，驱动程序不会尝试与物理设备进行通信，但服务器将继续返回有效的 OPC 数据。模拟停止与设备的物理通信，但允许 OPC 数据作为有效数据返回到 OPC 客户端。在“模拟模式”下，服务器将所有设备数据处理为反射型：无论向模拟设备写入什么内容，都会读取回来，而且

会单独处理每个 OPC 项。项的内存映射取决于组更新速率。如果服务器移除了项 (如服务器重新初始化时), 则不保存数据。默认值为“否”。

● **注意:**

1. “系统”标记 (`_Simulated`) 为只读且无法写入, 从而达到运行时保护的目。 “系统”标记允许从客户端监控此属性。
2. 在“模拟”模式下, 项的内存映射取决于客户端更新速率 (OPC 客户端的“组更新速率”或本机和 DDE 接口的扫描速率)。这意味着, 参考相同项、而采用不同更新速率的两个客户端会返回不同的数据。

● “模拟模式”仅用于测试和模拟目的。该模式永远不能用于生产环境。

设备属性 - 扫描模式

“扫描模式”为需要设备通信的标记指定订阅客户端请求的扫描速率。同步和异步设备的读取和写入会尽快处理; 不受“扫描模式”属性的影响。

属性组	☐ 扫描模式	
常规	扫描模式	遵循客户端指定的扫描速率
扫描模式	来自缓存的初始更新	禁用
定时		

“扫描模式”: 为发送到订阅客户端的更新指定在设备中扫描标记的方式。选项说明如下:

- “遵循客户端指定的扫描速率”: 此模式可使用客户端请求的扫描速率。
- “不超过扫描速率请求数据”: 此模式可将该数值集指定为最大扫描速率。有效范围为 10 至 99999990 毫秒。默认值为 1000 毫秒。
● **注意:** 当服务器有活动的客户端和设备项且扫描速率值有所提高时, 更改会立即生效。当扫描速率值减小时, 只有所有客户端应用程序都断开连接, 更改才会生效。
- “以扫描速率请求所有数据”: 此模式将以订阅客户端的指定速率强制扫描标记。有效范围为 10 至 99999990 毫秒。默认值为 1000 毫秒。
- “不扫描, 仅按需求轮询”: 此模式不会定期轮询属于设备的标签, 也不会在一个项变为活动状态后为获得项的初始值而执行读取操作。客户端负责轮询以便更新, 方法为写入 `_DemandPoll` 标记或为各项发出显式设备读取。有关详细信息, 请参阅服务器帮助中的“设备需求轮询”。
- “遵循标签指定的扫描速率”: 此模式将以静态配置标记属性中指定的速率强制扫描静态标记。以客户端指定的扫描速率扫描动态标记。

“来自缓存的初始更新”: 启用后, 此选项允许服务器为存储 (缓存) 数据的新激活标签参考提供第一批更新。只有新项参考共用相同的地址、扫描速率、数据类型、客户端访问和缩放属性时, 才能提供缓存更新。设备读取仅用于第一个客户端参考的初始更新。默认设置为禁用; 只要客户端激活标记参考, 服务器就会尝试从设备读取初始值。

设备属性 - 标记生成

自动生成标记数据库功能可使设置应用程序成为一项即插即用操作。选择可以配置为自动构建标记列表的通信驱动程序 (标记与特定于设备的数据相对应)。可以从客户端浏览这些自动生成的标记 (这取决于支持驱动程序的性质)。

● 并非所有设备和驱动程序都支持全自动生成标记数据库, 也并非所有都支持相同的数据类型。有关具体信息, 请参阅各驱动程序的数据类型说明或支持的数据类型列表。

如果目标设备支持其自身的本地标记数据库, 则驱动程序会读取设备的标记信息, 并使用该数据来在服务器中生成标记。如果该设备本身不支持已命名的标记, 则驱动程序会根据特定于驱动程序的信息来创建标记列表。这两个条件的示例如下:

1. 如果数据采集系统支持其自身的本地标记数据库, 则通信驱动程序将使用在设备中发现的标记名称来构建服务器的标记。
2. 如果以太网 I/O 系统支持其自身可用 I/O 模块类型的检测, 则通信驱动程序会基于插入以太网 I/O 机架的 I/O 模块类型在服务器中自动生成标记。

● **注意:** 自动生成标记数据库的操作模式可进行完全配置。有关详细信息, 请参阅下方的属性说明。

属性组	☐ 标记生成	
常规	设备启动时	启动时不生成
扫描模式	对于重复标记	创建时删除
定时	父组	
自动降级	允许自动生成的子组	启用
标记生成	创建	创建标记
冗余		

“属性更改时”: 如果设备支持在特定属性更改时自动生成标记, 系统会显示**“属性更改时”**选项。默认情况下, 该选项设置为**“是”**, 但可以将其设置为**“否”**来控制何时生成标记。在此情况下, 必须手动执行**“创建标记”**操作以执行标记生成。

“设备启动时”: 指定自动生成 OPC 标记的时间。选项说明如下:

- **“启动时不生成”**: 此选项可防止驱动程序向服务器的标记空间添加任何 OPC 标记。这是默认设置。
- **“始终在启动时生成”**: 此选项可使驱动程序评估设备, 以便获得标记信息。每次启动服务器时, 它还会向服务器的标记空间添加标记。
- **“首次启动时生成”**: 此选项可使驱动程序在首次运行项目时评估目标设备, 以便获得标记信息。它还可以根据需要向服务器标记空间添加任何 OPC 标记。

● **注意**: 如果选择自动生成 OPC 标记的选项, 添加到服务器标记空间的任何标记都必须随项目保存。用户可以在**“工具”|“选项”**菜单中将项目配置为自动保存。

“对于重复标记”: 启用自动生成标记数据库后, 服务器需要了解如何处理先前已添加的标记, 或在初始创建通信驱动程序后添加或修改的标记。此设置可控制服务器处理自动生成的以及当前存在于项目中的 OPC 标记的方式。它还可以防止自动生成的标记在服务器中累积。

例如, 如果用户更改机架中的 I/O 模块, 并且服务器配置为**“始终在启动时生成”**, 则每当通信驱动程序检测到新的 I/O 模块时, 新标记就会添加到服务器。如果未移除旧标记, 则许多未使用的标记可能会在服务器的标记空间中累积。选项包括:

- **“创建时删除”**: 此选项可在添加任何新标记之前, 将先前添加到标记空间的任何标记删除。这是默认设置。
- **“根据需要覆盖”**: 此选项可以指示服务器仅移除通信驱动程序要用新标记替换掉的标记。所有未被覆盖的标记仍将保留在服务器的标记空间中。
- **“不覆盖”**: 此选项可以防止服务器移除任何之前生成的标记或服务器中已存在的标记。通信驱动程序只能添加全新的标记。
- **“不覆盖, 记录错误”**: 此选项与前一选项有相同效果, 并且在发生标记覆盖时, 也会将错误消息发布到服务器的事件日志。

● **注意**: 删除 OPC 标记会影响通信驱动程序已自动生成的标记以及使用匹配已生成标记的名称添加的任何标记。如果标记所使用的名称可能与驱动程序自动生成的标记相匹配, 则用户应避免将此类标记添加到服务器。

“父组”: 此属性通过指定将要用于自动生成标记的组, 来防止自动生成的标记与已手动输入的标记发生混淆。组名称最多可包含 256 个字符。此父组具有一个根分支, 可将所有自动生成的标记添加到其中。

“允许自动生成的子组”: 此属性用于控制服务器是否为自动生成的标记自动创建子组。这是默认设置。如果禁用, 则服务器会在没有任何分组的简单列表中生成设备标记。在服务器项目中, 生成的标记使用地址值命名。例如, 生成过程中不会保留标记名称。

● **注意**: 如果在服务器生成标记的过程中, 分配给标记的名称与现有标记的名称相同, 则系统会自动递增到下一个最高数字, 以免标记名称发生重复。例如, 如果生成过程中创建了名为 **“AI22”** 的标记且该名称已存在, 则会将标记创建为 **“AI23”**。

“创建”: 开始创建自动生成的 OPC 标记。如果已修改设备的配置, 则**“创建标记”**可强制驱动程序重新评估设备以发现可能的标记更改。由于该选项可以通过系统标记进行访问, 这使得客户端应用程序能够启动标记数据库创建。

● **注意**: 当**“配置”**对项目进行离线编辑时, 会禁用**“创建标记”**。

设备属性 - Configuration API

以下属性使用 Configuration API 服务来定义设备。

常规属性

```
common.ALLTYPES_NAME * 必需的参数
common.ALLTYPES_DESCRIPTION
servermain.DEVICE_CHANNEL_ASSIGNMENT
servermain.MULTIPLE_TYPES_DEVICE_DRIVER * 必需的参数
servermain.DEVICE_MODEL * 并非必需参数，但应验证默认值是否可接受
servermain.DEVICE_ID_STRING * 必需的参数
servermain.DEVICE_DATA_COLLECTION
servermain.DEVICE_SIMULATED
```

扫描模式

```
servermain.DEVICE_SCAN_MODE
servermain.DEVICE_SCAN_MODE_RATE_MS
servermain.DEVICE_SCAN_MODE_RATE_MS
servermain.DEVICE_SCAN_MODE_PROVIDE_INITIAL_UPDATES_FROM_CACHE
```

自动降级

```
servermain.DEVICE_AUTO_DEMOTION_ENABLE_ON_COMMUNICATIONS_FAILURES
servermain.DEVICE_AUTO_DEMOTION_DEMOTE_AFTER_SUCESSIVE_TIMEOUTS
servermain.DEVICE_AUTO_DEMOTION_PERIOD_MS
servermain.DEVICE_AUTO_DEMOTION_DISCARD_WRITES
```

标记生成

```
servermain.DEVICE_TAG_GENERATION_ON_STARTUP
servermain.DEVICE_TAG_GENERATION_DUPLICATE_HANDLING
servermain.DEVICE_TAG_GENERATION_GROUP
servermain.DEVICE_TAG_GENERATION_ALLOW_SUB_GROUPS
```

 **提示:** 要调用“自动生成标记”，请向设备上的 TagGeneration 服务端点发送正文为空的 PUT。
 **另请参阅:** 有关详细信息，请参阅“服务”帮助。

定时

```
servermain.DEVICE_CONNECTION_TIMEOUT_SECONDS
servermain.DEVICE_REQUEST_TIMEOUT_MILLISECONDS
servermain.DEVICE_RETRY_ATTEMPTS
servermain.DEVICE_INTER_REQUEST_DELAY_MILLISECONDS
```

● **另请参阅:** 服务器帮助系统“*Configuration API 服务*”部分。

数据类型说明

必须为每一个可访问的地址分配一个数据类型。Simulator 设备支持以下数据类型。

数据类型	说明
BCD	两个字节封装的 BCD 值的范围是 0 至 9999。对于超出此范围的值，未定义行为。
布尔型	单个位
字节	无符号 8 位值 位 0 是低位 位 7 是高位
字符	有符号 8 位值 位 0 是低位 位 6 是高位 位 7 是符号位
日期	浮点 OLE 自动日期 (映射到 VARIANT VT_DATE 数据类型)。
双精度*	64 位浮点值 驱动程序将四个连续寄存器解释为双精度值，方法将最后两个寄存器作为高位双字，将前两个寄存器作为低位双字。
双精度示例	如果将寄存器 K0001 指定为双精度数，则寄存器 K0001 的位 0 将是 64 位数据类型的位 0，寄存器 K0004 的位 15 将是 64 位数据类型的位 63。
双字型	无符号 32 位值 位 0 是低位 位 31 是高位
浮点型*	32 位浮点值 驱动程序将两个连续寄存器解释为单精度值，方法是将最后一个寄存器作为高位字，将第一个寄存器作为低位字。
浮点型示例	如果将寄存器 K0001 指定为浮点型，则寄存器 K0001 的位 0 将是 32 位数据类型的位 0，寄存器 K0002 的位 15 将是 32 位数据类型的位 31。
LBCD	四个字节封装的 BCD 值的范围是 0 至 99999999。对于超出此范围的值，未定义行为。
长整型	有符号 32 位值 位 0 是低位 位 30 是高位 位 31 是符号位
LLong	有符号 64 位值 位 0 是低位 位 62 是高位 位 63 是符号位
四字型	无符号 64 位值 位 0 是低位 位 63 是高位
短整型	有符号 16 位值 位 0 是低位 位 14 是高位 位 15 是符号位
字符串	空终止 ASCII 字符数组
字	无符号 16 位值 位 0 是低位 位 15 是高位

*说明假定为默认；即，按首双字排序为低处理 64 位数据类型，按首字排序为低处理 32 位数据类型。

地址说明

Simulator 驱动程序支持四种类型的地址：B 寄存器、R 寄存器、K 寄存器和 S 寄存器。R 和 K 寄存器为数字数据。S 寄存器是可变长度字符串数据位置。

B 寄存器模拟线圈，并且是真正的布尔型。布尔型寄存器不会在扫描时自动更改值。当 R 寄存器作为字符、字节、字、短整型、BCD、长整型、双字型、双长整型、四字型或 LBCD 引用时，它将通过每次读取时递增 1 的方式来模拟不断更改的数据。这些类型的数组在每次读取时递增 1。当 R 寄存器作为浮点型或双精度型引用时，值在每次读取时递增 1.25。浮点型或双精度型的数组在每次读取时递增 1.25。每种类型均有增量范围。对于浮点型，范围为 0 至 32767。对于双精度型，范围为 0 至 65535。

K 和 R 寄存器的初始值均为零。S 寄存器的初始值为“字符串数据 Sn”，其中 n 是寄存器编号。

地址范围和数据类型规范因使用的型号而异。Simulator 驱动程序还支持新的模拟函数，其中包括斜坡函数、正弦函数、随机函数和用户定义函数。有关详细信息，请从下表中选择一个链接。

[8 位设备地址](#)

[16 位设备地址](#)

8 位设备地址

8 位设备的内存配置被模拟为一块编号 0 到 9999 的字节位置。字节的寻址方式为相对于块开始的偏移。每个格式的默认数据类型以**粗体**显示。

设备类型	范围	数据类型	访问
寄存器	R0000-R9999 R0000-R9998 R0000-R9996 R0000-R9992	字节 、字符 字、短整型、BCD、 双字型、长整型、LBCD、浮点型、 LLong、QWord、双字型、 日期、布尔值	读/写
常量	K0000-K9999 K0000-K9998 K0000-K9996 K0000-K9992	字节 、字符 字、短整型、BCD 双字型、长整型、LBCD、浮点型 LLong、QWord、双字型、 日期、布尔值	读/写
位	R0000.0-R9999.7 K0000.0-K9999.7	布尔型	读/写
布尔型	B0000-B9999	布尔型	读/写
字符串	S000-S999	字符串	读/写

注意：

- 除位级布尔值之外，所有数据类型均支持数组 (将 [r] 或 [r][c] 附加到地址后面)。
- 最大数组大小基于地址类型的上部范围。例如，具有字节数据类型的 R 寄存器数组的最大大小为 1000 个元素。示例：R0000[1000]，字数组的最大大小，R0000 [500]。
- 为数据类型指定的地址必须满足数据类型的完整大小。这意味着用户的写操作不能超出数据范围。

另请参阅：[模拟函数](#)

16 位设备地址

16 位设备的内存配置被模拟为一块编号 0 到 9999 的字位置。字的寻址方式为相对于块开始的偏移。每个格式的默认数据类型以**粗体**显示。

设备类型	范围	数据类型	访问
寄存器	R0000-R9999 R0000-R9998 R0000-R9996	字 、短整型、BCD 双字型、长整型、LBCD、浮点型 LLong、QWord、双字型、 日期、布尔值	读/写

设备类型	范围	数据类型	访问
常量	K0000-K9999 K0000-K9998 K0000-K9996	字、短整型、BCD 双字型、长整型、LBCD、浮点型 LLong、QWord、双字型、 日期、布尔值	读/写
位	R0000.00-R9999.15 K0000.00-K9999.15	布尔型	读/写
布尔型	B0000-B9999	布尔型	读/写
字符串	S000-S999	字符串	读/写

● **注意：**

1. 除位级布尔值之外，所有数据类型均支持数组 (将 [r] 或 [r][c] 附加到地址后面)。
2. 最大数组大小基于地址类型的上部范围。例如，具有字数据类型的 R 寄存器数组的最大大小为 1000 个元素。示例：R0000[1000]，字数组的最大大小，R0000 [500]。
3. 为数据类型指定的地址必须满足数据类型的完整大小。这意味着用户的写操作不能超出数据范围。

● **另请参阅：** [模拟函数](#)

模拟函数

模拟函数可用于创建模拟许多真实数据源的 OPC 项。每一个模拟函数均提供了不同的输出，这些输出具有多个常见属性，例如**速率**、**下限**和**上限**。“速率”(也称为更改率)可用于指定模拟值更改状态的频率。速率值以毫秒计数的形式输入，其范围介于 10 至 3600000 之间。此更改率完全独立于客户端应用程序读取数据的频率。与 PLC 类似，模拟函数始终在后台运行。

“下限”和“上限”

“下限”和“上限”属性可用于指定由模拟函数生成的数据的范围。通过使用“下限”和“上限”属性，用户可以生成模拟值，只需调整数据范围即可对这些模拟值进行偏移。对于支持限制的模拟函数，范围的输入格式可用于确定模拟值的数据类型。例如，输入斜坡标记时，如果下限为 75，上限为 3000，则生成的 OPC 项会被视为长整型数据类型。输入相同的斜坡标记时，如果下限为 75.123，上限为 3000.567，则生成的 OPC 项会被视为浮点型数据类型。在这两个示例中，默认的数据格式为浮点型或长整型，但可选择任何可用的数据类型作为任何模拟函数的输出格式。但是，由下限和上限指定的范围必须与所需的数据类型选项相匹配。

与上述基于寄存器的地址不同，可输入的模拟函数数目并没有限制。每个具有唯一属性的模拟函数将被视为新的模拟值。因此，为了在客户端应用程序中读取多个位置的相同值，我们需要创建寻址模拟函数，这时，在每种情况下采用相同方式输入模拟函数至关重要。

模拟函数为只读对象

模拟函数为只读对象。一旦客户端应用程序开始从模拟函数中读取数据，该函数即会继续生成数据，直到客户端应用程序将项删除为止。输入浮点属性时，模拟函数不接受指数形式的数字值条目。

函数定义

[斜坡函数](#)

[随机函数](#)

[正弦函数](#)

[用户函数](#)

斜坡函数

RAMP(速率,下限,上限,增量)

斜坡函数用于创建指定数字范围内递增或递减的值。应使用下限和上限设置所需的范围。可调整下限或上限，以对生成的数据应用偏移。增量值可为正数或负数。如果增量值为正，则值将以所需速率生成由下限到上限的斜坡。如果增量值为负，则值将以所需速率生成由上限到下限的斜坡。输入的下限、上限和增量值可为整数或浮点格式。

支持的数据类型

字节、字符、字、四字型、双字型、短整型、**长整型**、双长整型、浮点型、双精度

示例

RAMP(120, 35, 100, 4)

将创建增量为每 120 毫秒递增 4 的 35 到 100 的值。

RAMP(300, 150.75, 200.50, -0.25)

将创建增量为每 300 毫秒递减 0.25 的 200.50 到 150.75 的值。

随机函数

RANDOM(速率, 下限, 上限)

斜坡函数用于创建指定数字范围内可随意更改的项。应使用下限和上限设置所需的范围。可调整下限或上限，以对生成的数据应用偏移。

支持的数据类型

Byte, Char, Short, Word, DWord, QWord, **Long**, LLong, Float, Double

示例

RANDOM(30, -20, 75)

将创建速率为 30 毫秒的从 -20 到 75 随机变化的值。

正弦函数

正弦 (“速率”、“下限”、“上限”、“频率”、“相位”)

正弦函数可用于创建遵循正弦值更改的项。应使用下限和上限设置所需的范围。可调整下限或上限，以对生成的数据应用偏移。“频率”属性可用于指定所需的波形 (以赫兹为单位)。最大的有效频率大约为 5 赫兹。“频率”属性的有效范围为 0.001 至 5 赫兹。“相位”属性可用于偏移由特定角度生成的正弦波。输入“相位”的范围应在 0.0 到 360.0 之间。在这种情况下，对于模拟函数的运行方式，“速率”属性起到了关键作用。为获得此函数的良好正弦输出，“速率”必须至少比所需“频率”快两倍。例如，对于在 200 毫秒的速率下发生变化的正弦波，如果用户想要得到 5 赫兹的正弦波，则“速率”属性的最大值应设置为 100 毫秒。为了获得最佳的正弦波结果，建议将“速率”设置为 10 或 20 毫秒。正弦函数“速率”的有效范围为 10-1000 毫秒。

支持的数据类型

浮点型、双精度

示例

正弦 (10, -40, 40, 2, 0)

这将创建频率为 2 赫兹的正弦值，其范围介于 -40 至 40 之间，并且不具有相位偏移。

User 函数

USER(速率, 用户值 1, 用户值 2, 用户值 3, ...用户值 N)

USER 函数为定义模拟函数返回的数据类型提供了最大的灵活性。与其他函数在指定范围内进行操作有所不同，USER 函数可用于指定一组数字或字符串值以指定速率进行循环。输入的值用于确定项目的数据类型。例如，如果输入 100.45 作为用户值之一，模拟对象的输出将被视为浮点数据。如果其中一个输入的用户值是 "Hello World"，则模拟对象的输出将被视为字符串数据。定义项目后，可通过指定所需的数据类型来覆盖这些默认选择。

● **注意:** 当输入用户值为字符串时，可以通过在字符串前添加反斜杠 "\" 来在字符串中输入逗号。

支持的数据类型

Bool, Byte, Char, Short, Word, DWord, QWord, Long, LLong, Float, Double

● **注意:** USER 模拟函数中输入的值用于确定默认数据类型。

示例

USER(250, Hello, World, this, is, a, test)

这会创建以 250 毫秒速率从序列中的一个文本自动更改为下一个的字符串数据类型值。

USER(20, 1.25, 100.56, 200.11, 75.1)

这会创建以 20 毫秒速率从序列中的一个浮点值更改为下一个的浮点数据类型值。

USER(50, 1,1,0,1,0,1,0,0,1,1,1,0,0,0)

这会生成以 50 毫秒速率从序列中的一个布尔状态更改为下一个的布尔型值。这可用于创建非常复杂的位阵列。

USER(1000, In this case\, , I needed to use a \, in , my text)

需要在这些文本片段中添加 "\", 以便在文本值中插入一个逗号。此外, 如有需要, 每个值的文本可以是句子或句子片段。

事件日志消息

以下信息涉及发布到主要用户界面中“事件日志”窗格的消息。。请参阅有关筛选和排序“事件日志”详细信息视图的服务器帮助。服务器帮助包含许多常见的消息，因此也应对其进行搜索。通常，其中会尽可能提供消息的类型 (信息、警告) 和故障排除信息。

无法加载项状态数据。原因:<原因>。

错误类型：

警告

可能的原因：

1. 驱动程序因特定原因无法加载项状态数据。
2. 数据文件损坏。
3. 磁盘空间不足。
4. 路径中的驱动无效。
5. 数据文件已删除或重命名。

可能的解决方案：

解决方案取决于错误消息中给出的原因。如果文件损坏或已删除，则之前的状态数据将丢失。

无法保存项状态数据。原因:<原因>。

错误类型：

警告

可能的原因：

1. 驱动程序因特定原因无法保存项状态数据。
2. 数据文件损坏。
3. 磁盘空间不足。
4. 路径中的驱动无效。
5. 数据文件已删除或重命名。

可能的解决方案：

解决方案取决于错误消息中给出的原因。如果文件损坏或已删除，则之前的状态数据将丢失。

索引

1

16 位设备地址 15

8

8 位设备地址 15

B

BCD 13

I

ID 8

L

LBCD 13

LLong 13

U

User 函数 17

帮

帮助内容 3

标

标记生成 9

标识 8

不

不扫描, 仅按需求轮询 9

布

布尔型 13

操

操作模式 8

常

常规 8

创

创建 10

存

存在性 5

地

地址说明 15

短

短整型 13

对

对于重复标记 10

浮

浮点型 13

父

父组 10

覆

覆盖 10

概

概述 3

寄

寄存器 15

来

来自缓存的初始更新 9

名

名称 8

模

模拟 9

模拟函数 16

驱

驱动程序 8

日

日期 13

扫

扫描模式 9

删

删除 10

设

设备启动时 10

设备属性 - 标记生成 9

设置 3

生

生成 10

事

事件日志消息 19

属

属性更改时 10

数

数据类型说明 13

数据收集 8

双

双精度 13

双字型 13

四

四字型 13

随

随机函数 17

通

通道分配 8

无

无法保存项状态数据。原因
<原因>。19

无法加载项状态数据。原因
<原因>。19

斜

斜坡函数 16

型

型号 8

允

允许子组 10

长

长整型 13

正

正弦函数 17

字

字 13

字符 13

字符串 13

字节 13

遵

遵循标签指定的扫描速率 9